Amendments to the Drawings:

Please substitute the attached Replacement Sheets of drawings for all of the drawings now on file (13 sheets):

Figs. 1A-1D, 2, 3A-3C, 4-10, 11A-11E, 12A and 12B.

The additions/changes of the numeric designations are as follows:

Fig. 3B: 21 and 23 have been added.

Fig. 3C: 22, 22' and 31 have been added.

Fig. 4: 40, 25, 26 and 27 have been added.

REMARKS/ARGUMENTS

Claims 5 - 7, 11 - 13, 16, 17 and 19 are cancelled without prejudice.

The specification and drawings have been amended to attend to the objections noted by the Examiner on pages 2 and 3 of the Office Action.

Claims 5 - 7, 11 - 13, 16, 17 and 19 have been cancelled. Claims 1, 2, 3, 4, 8, 9, 10, 14, 15 and 18 remain in the application. These claims have been amended to avoid the claim objections noted by the Examiner, the 35 U.S.C. 112 claim rejections noted by the Examiner on pages 4 and 5 of the Office Action and the claim rejections under 35 U.S.C. 101 noted at pages 5, 6 and 7 of the Office Action. Claim 4 has been amended to cover the subject matter of claim 19, and claim 19 has been cancelled.

The rejection of claims 1, 2 and 3 under 35 U.S.C. 102(b) as being anticipated by Herbert et al, Southeast Symposium on System Theory, March 18-20, 2001, Proceedings of the 33rd Southeastern Symposium on System Theory, pp. 315-318 (hereinafter Herbert) is respectfully traversed.

Herbert presents how to use an existing software tool (IPCHAINS, in the Linux kernel) for NAT, while the present invention provides a data structure which could be used for instance to <u>implement</u> a tool such an IPCHAINS.

In network nodes (routers, bridges, etc.) which are not using Linux, the filtering is also rule-based, and is usually called ACL

(access control list). In a Linux box, ACLs could be for instance declared using IPCHAINS.

However, the key question is: how to design efficient storage and retrieval methods for these rules. Herbert does not describe the actual software code for IPCHAINS, but only its usage (external interface).

Existing methods for implementing rules storage and retrieval range from fast but expensive hardware-based solutions (especially TCAMs) to cheap but inefficient software-based solutions (the worst could be simple linked lists, or database techniques using hashing, or prefix-based solutions, or the best known state of the art FIS: Fat Inverted Segment, or HiCuts, a heuristic solution with a large preprocessing time).

The present invention provides a good hybrid method (software-based, but faster than any known software-based method) for multi-field classification. Unlike FIS which requires multiple traversals to find the matching rule, applicants' method requires only one traversal, which is why it is faster. It also uses less memory footprint, and is faster to build at setup. While hardware solutions are even faster, they are definitely more expensive.

Even the external specification is different, since IPCHAINS can only specify rules one field at a time (not multi-field), and only unique values (no intervals, except for the IP subnets, which are less general than arbitrary integer intervals which we use as

input). Also, with IPCHAINS, only pre-programmed fields can be used for a rule.

A simple analogy follows: Rejecting the method of structuring data, as disclosed and claimed, because of Linux IPCHAINS, disclosed in Herbert would be like rejecting an idea for a new engine technology because the concepts of cars, lawnmowers, airplanes already exist. With a new improved engine, you will use less gas and pollute less, but you will still drive your car the same way. In other words, you will still have ACLs, filtering, NATs, and tools like IPCHAINS, but you will use more efficient data structures for storing and accessing the range-specified rules.

Thus, the rejection of claims 1 - 4 based on Herbert is not believed to be justified.

The rejection of claims 8, 10 and 18 under 35 U.S.C. 102(b) as being anticipated by Lakshman et al (US 6,341,130) (hereinafter Lakshman) is respectfully traversed.

Applicants' Claim 8 specifies *inter alia*:

> ...forming the tree structure such that each node of the tree contains a single elementary interval, an indication of original intervals associated with the elementary interval, and pointers to any adjacent nodes in the tree.

The Examiner refers to portions of the abstract of Lakshman which in context read as follows:

> Then, in the other dimension, the overlapping filter rectangle segments are decomposed into non-overlapping intervals, and the highest priority filter-rule overlapping each non-overlapping interval is associated with that interval. A filter-rule table is then constructed with entries ordered according to prefix length and non-overlapping interval, each entry

> associated with a particular filter-rule. A packet classification algorithm then matches the field or other parameter information in the packet to the filter-rule table entries to identify the filter-rule rectangle associated with the filter-rule to be applied to the packet.

Applicants respectfully submit that this does not appear to describe:

> ...the tree structure such as that each node of the tree contains a single elementary interval, an indication of original intervals associated with the elementary interval and pointers to any adjacent nodes in the tree.

Claims 9, 10 and 18 depend from claim 8 and are patentable for the same reason.

The rejections of claims 8, 9, 14 and 15 under 35 U.S.C. 102(e) as being anticipated by Henderson et al (US PG Pub. No. 2004/0133590) (hereinafter Henderson) is respectfully traversed.

Claim 8 and its dependent claims specify:

> ...forming the tree structure such that each node of the tree contains a single elementary interval, an indication of original intervals associated with the elementary interval, and pointers to any adjacent nodes in the tree.

The Examiner refers to paragraph 0066 and step 210 of Fig. 2 of Henderson and Fig. 1d, 1b and paragraph 0062 and step 355. As stated in paragraph 0067 of Henderson:

> ...As indicated in block 210, data items for overlapping ranges may be fragmented into multiple keys.

Note in paragraph 0066, Henderson states:

> ...in general, after completion of the appropriate insertion process, keys are arranged sequentially.

This is not the same or equivalent of applicants':

> ...step of forming the tree structure such as that each node of the tree contains a single elementary interval, an indication of original intervals associated with the elementary interval and pointers to any adjacent nodes in the tree.

Referring now to applicants' claim 14, this claim has been amended to place it a context of a computer-based communication system. Moreover, the claim calls for providing a disjoint intervals tree from a range specified rule set, each rule in the rule set having an equal number of fields and each field specifying a range having an upper and lower bound forming a set of intervals, the method comprising:

> combining overlapping intervals of the set of intervals to form larger intervals that are <u>disjoint</u> to each other; and
>
> finding the maximum disjoint intervals for the set of intervals.

Clearly, this is not taught in paragraph 0050 of Henderson, nor is it augmented by paragraphs 0089 - 0092 and paragraph 0102. The word "disjoint" as used in applicants' claim 14 or its equivalent does not appear in the paragraph referred to by the Examiner. In fact, it is nowhere in the Henderson specification, claims or drawings.

The rejection of claim 1 under 35 U.S.C. 103(a) as being unpatentable over Henderson et al (US PG Pub. No. 2004/0133590) (hereinafter Henderson) in view of Donald Knuth's Fundamental Algorithms, Vol. 1, 2nd edition, Addison-Wesley, 1973, p. 316-317 (hereinafter Knuth) is respectfully traversed.

Claim 1 calls for:

> forming a first layer of the tree-like data structure made up of a set of non-overlapping integer intervals; and
>
> forming one or more additional layers, each additional layer being made up of a set of non-overlapping integer intervals and a set of overlapping integer intervals to provide said tree-like data structure;
>
> wherein the traffic flow evaluations can be carried out by one pass through the tree-like data structure.

Even though the Knuth reference teaches single-pass examination of nodes, it is not at all clear that traffic flow evaluation can be carried out by one pass through the tree-like data structure of Henderson.

Claim 10 stands rejected under 35 U.S.C. 103(a) as being applied to claim 8 as being unpatentable by a further reading of Henderson in view of Gallo (US 6,700,883).

Claim 10 depends from claim 8 and is patentable for the same reason.

Claims 7 - 13 have been cancelled.

In view of the above, further and favorable reconsideration is respectfully requested.

Respectfully submitted,

Jim Zegeer

Jim Zegeer, Reg. No. 18,957
Attorney for Applicants

Attachments: Replacement Sheets of Drawings (13 sheets)

Suite 108
801 North Pitt Street
Alexandria, VA 22314
Telephone: 703-684-8333

Date: September 8, 2006

In the event this paper is deemed not timely filed, the applicant hereby petitions for an appropriate extension of time. The fee for this extension may be charged to Deposit Account No. 26-0090 along with any other additional fees which may be required with respect to this paper.